Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: https://orca.cardiff.ac.uk/id/eprint/83175/

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Awan, Malik Shahzad Kaleem, Burnap, Peter ORCID: https://orcid.org/0000-0003-0396-633X and Rana, Omer Farooq ORCID: https://orcid.org/0000-0003-3597-2646 2016. Identifying cyber risk hotspots: A framework for measuring temporal variance in computer network risk. Computers and Security 57 , pp. 31-46. 10.1016/j.cose.2015.11.003 file

Publishers page: http://dx.doi.org/10.1016/j.cose.2015.11.003 <http://dx.doi.org/10.1016/j.cose.2015.11.003>

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

information services
gwasanaethau gwybodaeth

COMPUTERS & SECURITY 57 (2016) 31–46

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Identifying cyber risk hotspots: A framework for measuring temporal variance in computer network risk

Malik Shahzad Kaleem Awan *, *Pete Burnap*, *Omer Rana*

School of Computer Science and Informatics, Cardiff University, UK

ARTICLE INFO

Article history:
Received 4 June 2015
Received in revised form 18 September 2015
Accepted 4 November 2015
Available online 28 November 2015

Keywords:
Cyber attacks
Network traffic analysis
System risk
Risk score
Risk grade
Cyber hotspots
Risk assessment framework

ABSTRACT

Modern computer networks generate significant volume of behavioural system logs on a daily basis. Such networks comprise many computers with Internet connectivity, and many users who access the Web and utilise Cloud services make use of numerous devices connected to the network on an ad-hoc basis. Measuring the risk of cyber attacks and identifying the most recent modus-operandi of cyber criminals on large computer networks can be difficult due to the wide range of services and applications running within the network, the multiple vulnerabilities associated with each application, the severity associated with each vulnerability, and the ever-changing attack vector of cyber criminals. In this paper we propose a framework to represent these features, enabling real-time network enumeration and traffic analysis to be carried out, in order to produce quantified measures of risk at specific points in time. We validate the approach using data from a University network, with a data collection consisting of 462,787 instances representing threats measured over a 144 hour period. Our analysis can be generalised to a variety of other contexts.

© 2016 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Quantifying the risk of cyber attacks due to software applications on a computer network at any given time, measuring the impact of an attack, and understanding attack patterns are complex and challenging tasks. The complexity is compounded by an increasing number of networked devices being brought into modern networked environments on an ad-hoc basis. Such devices often have a large number of software applications ("apps") installed, which can leave the devices, and subsequently the whole network, vulnerable to cyber threats. To compound the cyber security management difficulties posed by such infrastructure, the techniques and attack vectors used by cyber criminals are increasingly mature, stealthy and dynamic. Installing and configuring network defence systems such as Firewalls and Intrusion Detection Systems (IDS) provide an initial barrier to cyber attacks but the inclusion of personal devices and remote working has contributed to the de-perimeterisation of computer networks (Burnap and Hilton, 2009), meaning that network level defences alone are not sufficient to defend against these new and emerging issues. Individuals, companies, and governments may become victims of cyber crime, as well as (involuntary) helpers of cyber criminals.

Advances in network traffic analysis (Dantu et al., 2004; Frigault et al., 2008; Liu and Man, 2005; Poolsappasit et al., 2012; Xie et al., 2010) have enabled network administrators to track and respond to malicious activity with increasing levels of accuracy and effectiveness to reduce the potential for harm on network users. However, given the large number of applications

* *Corresponding author.* Tel.: +442920874812.
E-mail address: AwanMS@cardiff.ac.uk (M.S.K. Awan).
http://dx.doi.org/10.1016/j.cose.2015.11.003
0167-4048/© 2016 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

running within each network, the vulnerabilities inherent in these applications with new ones being discovered by cyber criminals on a daily basis, and the variance in the severity of an exploit or a vulnerability will affect the network and users. Therefore, conducting real-time risk assessments and determining the most pertinent risks to manage at any given time is far from straightforward. A need arises for an effective, cost-efficient and reliable mechanism for continuously monitoring and assessing risk arising due to software applications.

The continuous monitoring and assessment of risk mainly depends on an automated data collection process for storing logs containing malicious traffic instances and then their analysis. Such an analysis could help identify cyber-attack patterns, the associated risks with different types of cyber-attacks and their assessment in terms of some numeric score to help a network administrator take appropriate responsive actions (Kott and Arnold, 2013). This whole process involves two main challenges: (i) integration and pre-processing of the data feeds with diverse content and timeliness, originating across different segments of an organisational network; and (ii) using a suitable risk scoring algorithm to help support a network administrator better characterise risk under given environmental conditions. However, existing risk scoring and assessment mechanisms such as Common Vulnerability Scoring System (CVSS) (First, 2015), CORAS (Lund et al., 2011), ISO/IEC 27001:2013 (ISO, 2013), etc., require subjective, human-based qualitative inputs, which could potentially lead to inaccurate (ambiguous) and inconclusive results. Further, presenting a holistic view of the organisation-wide risk, distributed over different segments of the network over time, is of much more importance than simply reporting static risk scores (generally achieved by simply summing up the reported vulnerabilities across the network and reporting this as a single numeric value) (Kott and Arnold, 2013).

In this paper we propose a framework for the formalisation and integration of risk factors affecting a computer network (primarily a University network). Such a framework enables risk scores to be produced at various levels within the network over time, underpinning a dynamic probabilistic risk assessment model, which are then combined to represent the relative risks within a network in much the same way as a geographic map of terrestrial crime would represent crime "hotspots". This provides an intuitive representation of risk and enables administrators to "zoom" in and out of the risk model to determine and understand the most "at-risk" components within a network, sub-network and hosted application over time. We consider the threats targeting a given software application running within a network, their severity, and their frequency of occurrence over a given period of time to calculate an overall risk score, which is then used to provide a separate view of cyber risk hotspots posed by internally and externally hosted software applications. The application-specific cyber risk hotspots are further investigated for identification of most critical threats targeting different software applications using a decision tree structure to help network administrator(s) take immediate responses. The main contributions of our work include:

1. proposing risk metrics which could be calculated more objectively, rather than using subjective, human-based qualitative inputs to identify cyber risk hotspots in a computer network
2. modelling temporal risk behaviour of software applications and understanding the effectiveness of security policies
3. devising mechanism for alerting network administrator to take precautionary measures before the risk score significantly increases
4. identifying cyber risk hotspots emerging over a period of time in a computer network
5. investigating causes of emerging cyber risk hotspots associated with a particular software application

Using system log data (from an IPS/IDS) to inform a dynamically updated risk assessment, we validate the framework by using real-time system logs collected within a University network. The logs provide the frequency of threat exploits at an application level, and a severity score for each threat. A University network is particularly suitable to validate the framework as it contains multiple sub-networks distributed across different geographic locations, more than 34,000 users (both staff and students), and a large number of computing devices. Furthermore, users tend to work remotely and access machines within the network from external locations, requiring Web-facing ports to be opened. Although we have used a measurement based study to validate our approach, the results presented can be generalised to other similar University networks (which also have similar user communities, also making use of similar types of hosted applications).

2. Related work

A Bayesian network based risk management framework, called Bayesian Attack Graphs (BAG), has been proposed for assessing risk in a computer network thereby enabling system administrators to make decisions, e.g., apply a security patch, use firewall, disable the service, disconnect from Internet, in the operational environment. The model incorporated both the cause–consequence relationships and the likelihoods of exploiting them for estimating security risks for an organisation based on the Common Vulnerability Scoring System (CVSS) metrics (First, 2015). BAG has been used for better understanding the causal relationships between various network states and computing the likelihood of exploiting such relationships. The study further proposed and developed a genetic algorithm for optimising administrators' objectives for mitigating risks (Poolsappasit et al., 2012). However, the authors have only considered risk at a particular point in time, and not considered the likely variation in risk over time – the latter being an important component of our proposed risk assessment framework.

Dantu et al. (2004) used attack graphs to model network vulnerabilities and then deployed a Bayesian logic based probabilistic model for network risk assessment. They associated a probability with each of the nodes in the attack graph to represent the likelihood of an attack on a network. These probabilities were initially used for computing the likelihood of system compromise and later for calculating the organisational risk. They calculated organisational risk using synthetic network data. We

make use of real network data for identifying components within our framework and calculate the associated risk values.

Liu and Man (2005) modelled potential attack paths using Bayesian networks and considered attack mechanisms and background knowledge of attackers for computing an optimal subset of potential attack paths. The authors labelled their approach as a Bayesian attack graph. The nodes in Bayesian attack graph have an associated probability value describing the likelihood of an attack, which is then used for calculating organisational risk. They have not specified how to calculate the attack probabilities associated with each node. A security risk analysis of networked systems has been carried out using Bayesian networks to model uncertainties in attack structure, attacker actions and alerts. The study involved near real-time analysis of the network data to identify a suitable security response for the identified intrusions using a Bayesian network (Xie et al., 2010). The authors used the term Security Graph Model to refer to the graph models used for network security analysis. The study focused on identifying vulnerable machines, undetected exploits and missing threats. We have modelled attack patterns, their intensities and frequencies of occurrence under given conditions and have also considered temporal aspects of an attack and its spread through a network.

A dynamic Bayesian Network-based model, which incorporated the evolution of vulnerabilities over time in the attack graph, has been proposed by Frigault et al. (2008). The model derived its parameters from the widely used network values of CVSS (First, 2015) and the attack graphs, and focused on supporting security monitoring in a dynamic network environment. The authors used simulated examples for validating their model, whereas our model uses real-world data.

Dantu et al. (2007) have performed a study to estimate and classify attack behaviour types based on survey results. The study surveyed the social and behavioural profiles of individuals to classify their behavioural resources, e.g., skill level, time and attitude – as opportunist, hacker, or explorer behaviour. The classification was further used to determine the associated risk with the behavioural resources to formulate risk management strategies. Our framework models threats associated with individual software hosted over a network-based infrastructure, ranking the most vulnerable of these. Dantu et al. (2009) have hypothesised the relationship between network action sequence and attack behaviour, and proposed a 5-step model for detecting and estimating risk based on attack graphs and attacker behaviours. The model has been reported to be useful for minimising network vulnerabilities. Our framework models risks within a network over a particular time period and ranks the attack paths for consideration by a network administrator.

A service dependency graph based cost-sensitive intrusion-response system has been proposed to evaluate the three fundamental security principles: confidentiality, integrity and availability (Kheir et al., 2010). The study reported difficulties in identifying the potential impact on confidentiality and integrity after responding to an attack. Our framework uses parameters from the malicious network traffic data for calculating risk, which is then used for identifying the cyber risk hotspots in a computer network and assisting a network administrator to make suitable responses inline with the fundamental security principles.

A component-based architecture has been proposed for measuring the impact of an attack and then finding a suitable corresponding response while considering the associated costs and benefits. The model, being non-graph based in nature, uses an Observable Markov Decision Process to assess state of system assets from network security perspective (which can be normal, probing, under exploitation and compromised), selects a rational defence response for preserving security properties of a more secure system while considering potential cost factors (Zhang et al., 2009). Our framework evaluates and models risks within a network using objective parameters from the malicious network traffic data while following a continuous monitoring approach and helps in identifying and ranking the attack paths to enable a network administrator to take appropriate actions promptly.

A recent survey of risk assessment and network Intrusion Response Systems (IRS) proposed a taxonomy, highlighting their key features and limitations. The study suggested several research directions to improve IRS, one of which was to improve the link between risk assessment and intrusion response (Shameli-Sendi et al., 2014). Much of the existing literature attempts to automate this process. Our work presents an approach to risk assessment that provides a set of probabilistic metrics to represent risk at software application and sub-network levels, which can then be used to inform a security policy and lead to the dynamic configuration of intrusion responses. We have validated our proposed approach using real-world traffic data. Essentially, security policy can be modified by the network administrator based on the identification of emerging threats and risk "hotspots" in a network.

A hierarchical task network planning model using a non-graph based approach was proposed by Mu and Li (2010), which calculates risk thresholds by considering its positive and negative effects. The calculated risk threshold values have an associated response, which is triggered when a risk threshold exceeds a (pre-defined, static) response threshold. The approach reported less false positives due to the use of the response risk threshold. The study further proposed the use of a response selection window for supporting the most effective responses. We have used a decision tree for identifying and ranking attack paths to help a network administrator take appropriate actions. We do not make direct use of thresholds, however our approach can be easily extended with this. An important limitation is understanding what values these thresholds should be set to – often based on the experience of an administrator and the likely vulnerability of a network.

A dependency attack graph based approach, wherein the dependency attack graph is systematically integrated with Hidden Markov Models, has been proposed for investigating system-level vulnerabilities using probabilistic relations by Wang et al. (2013). The study suggested a set of organisation-level metrics and cost factors for calculating the associated cost of an attack and its defence. We assume that the cost of carrying out any action is identical, with the key selection criterion being the frequency of an attack.

An attack graph based model for evaluating risk in a considered computer network has been proposed by Kanoun et al. (2008), which computes risk by combining two major factors: potentiality and impact. Potentiality determines the probability of an attack being successful depending on natural

exposition and dissuasive measures, while impact parameters correspond to: (a) availability; (b) confidentiality; and (c) integrity and are dynamically calculated. We use objective parameters from the malicious network traffic for calculating risk. Our framework calculates risk values at different levels with software application threat as the fundamental cause of risk. A software application is susceptible to multiple threats while a subnetwork has many different types of software applications running in it. Further, our framework also supports continuous monitoring of risks and provides a consolidated view of risk at multiple levels in a computer network.

3. Risk assessment framework

Network administrators have the challenging responsibility of assessing risk to the networks at any time and then to plan the remedies accordingly. Traditionally, two methods for risk assessment are generally used – either the CORAS Method (Lund et al., 2011) and/or the use of a standard-based management framework such as ISO/IEC 27001:2013 (ISO, 2013). The CORAS Method (Lund et al., 2011) is a model-driven risk analysis approach that follows an 8-step methodology for threat and risk modelling. Structured brainstorming activities are performed for identifying threats, threat scenarios, vulnerabilities associated with organisational assets, risks, estimated impact of risk and suitable treatments for identified risks. CORAS Method suggests a one-off risk analysis of the system and as it involves meetings and lengthy discussions between various stakeholders for identifying risks and threat scenarios, it is not suitable for efficient and continuous monitoring of a computer network. Similarly, a management framework for risk assessment and their potential mitigation has been provided by ISO/IEC 27001:2013 (ISO, 2013), which requires continuous subjective input by the stakeholders for calculating potential risk to the considered network. Our risk assessment framework uses traffic logs without involving continuous subjective input at every time instance from a network administrator for assessing risk and can be effectively used for continuous risk monitoring at smaller time granularity. Further, the framework presents a holistic view of risk by first identifying cyber risk hotspots associated with a particular software application emerging over a period of time and then investigating the causes of their emergence.

A risk score is calculated based on the likelihood of a threat, and the business impact due to the threat represents the security risk assessment. The existing risk scoring algorithms have been driven by ad hoc heuristics, and are based on a subjective, human-based assessment. These often lead to risk metrics which have a potentially misleading and ambiguous nature (Kott and Arnold, 2013). Lack of theoretical foundations for quantitative characterisation of risk, its distribution over different segments of a network as well as time, and empirical validation of risk metrics require better mechanisms for computing and modelling risk across a computer network (Kott and Arnold, 2013). The risk score calculation has a further limitation of normalising different severity levels. Our risk assessment framework handles the limitations of both the CORAS Method and the ISO/IEC 27001:2013, as well as the existing (subjective) risk scoring algorithms. In this section we formalise the terms used in our proposed risk assessment framework and the associated network components to which they apply.

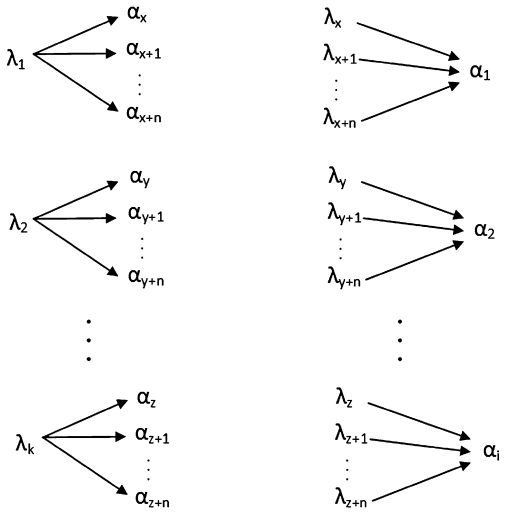
3.1. Main components

Our proposed risk assessment framework makes use of the following representation:

1. Software, $S = \{\alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_i\}$
2. Threat Category, $H = \{\Lambda_1, \Lambda_2, \Lambda_3, \cdots, \Lambda_j\}$
3. Threat, $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \cdots, \lambda_k\}$
4. Severity, $R = \{sev_1, sev_2, sev_3, \cdots, sev_l\}$

A computer network, divided into subnetworks Z, can host multiple software applications α within a subnet. A software application may be exposed to potential cyber attacks. We represent an attack as a threat, λ. An individual λ may be characterised using a number of parameters, e.g.: threat structure describing how an attack is carried out, delivery, breach and affect stages, targeted service, duration of attack. Threats can then be grouped based on common values across there parameters. The distinct j threat categories, Λ, are grouped in H. Each individual threat (λ) has associated effect/impact that it could inflict on a network. An inflicted effect by a threat has a certain level of severity (sev), with different severity levels that depend on the classification provided by either an intrusion detection system/firewall or the network administrator. The set, R, contains the defined l possible severity classifications for a given network environment. Once an attack has been identified, a suitable response is needed to control and minimise the damage due to the attack on a given subnet.

A network administrator can find a suitable response for a cyber attack by following either *Threat-focused* or *Software-focused* approach. *Threat-focused* approach identifies the most frequently occurring threats in a computer network with their multiple targeted software applications (see Fig. 1a). Such an approach requires iteratively fixing the effects of a particular



(a) Threat-focused Approach (b) Software-focused Approach

Fig. 1 – Cyber attack modelling approaches.

threat on each targeted software application, e.g., λ_1 targets $\alpha_x, \alpha_{x+1}, \cdots, \alpha_{x+n}$, λ_2 targets $\alpha_y, \alpha_{y+1}, \cdots, \alpha_{y+n}$ and λ_k targets $\alpha_z, \alpha_{z+1}, \cdots, \alpha_{z+n}$, then for each $\lambda_1, \lambda_2, \cdots, \lambda_k$ all the targeted software applications will require application of appropriate security controls by a network administrator. If $\alpha_x = \alpha_y = \alpha_z$ then this software application will be fixed in multiple iterations depending on the number of individual threats targeting it. From a network administrator's perspective, following a threat-focused approach for modelling cyber attacks and then finding and applying remedy controls for each threat independently can be a complex, lengthy and expensive process, however, the approach can be useful for cybersecurity researchers who are more interested in analysing the targets and behaviours of individual threats. On the other hand, *Software-focused* approach identifies most frequently targeted software applications due to multiple threats in a computer network (see Fig. 1b), e.g., α_1 is targeted by $\lambda_x, \lambda_{x+1}, \cdots, \lambda_{x+n}$, α_2 is targeted by $\lambda_y, \lambda_{y+1}, \cdots, \lambda_{y+n}$ and α_i is targeted by $\lambda_z, \lambda_{z+1}, \cdots, \lambda_{z+n}$. This is useful for identifying cyber hotspots due to software applications in a computer network from various views. Contrary to *Threat-focused* approach, a network administrator can easily take appropriate control measures for enhancing the security of the network by following Software-focused approach. We, in this study, take *Software-focused* approach for assessing and monitoring risk due to software applications running in a computer network.

We propose two risk metrics, Risk Score and Risk Grade, for identifying cyber risk hotspots in a computer network.

3.2. *Risk score*

Risk has been traditionally calculated as a product of likelihood of threats and impact on an organisation with both of the parameters depending on the subjective beliefs of the system administrator about the computer network of the organisation (Poolsappasit et al., 2012). Risk calculated in such a way has two main limitations: (1) using subjective beliefs of the system administrator; and (2) averaging out threat severities and their impact on the organisation. This is generally unhelpful, as understanding the potential impact of an individual threat is necessary to determine a potential response to it (rather than considering an average value that considers a variety of threats in aggregate).

We define *Risk* score associated with a particular threat as a product of: (1) conditional probability of a given software application targeted by threat(s); (2) severity of each threat instance; and (3) a constant factor to avoid averaging out varied threat severities for two more threat instances, summed over all software applications. A time unit *t* is associated with *Risk* score so that it can be used for continuous monitoring of software application risks across computer networks. We use *Risk* score for identifying cyber risk hotspots in a network to enable a network administrator get better insight of occurrence of potential risk patterns in different sub-networks, their trends and variations over a period of time and their sources across the network in much the same way as a terrestrial map of crime would represent crime at the country, county, policing ward and street level. A description of each of *Risk* score parameters is given below.

A software application α may offer multiple services. Each of these services may have an associated threat λ at time *t*. Hence, there may be multiple threats associated with the same application, making it possible to calculate the total vulnerability of an application by aggregating all threats at a particular point in time. The probability of a given software application, α, targeted by threat, λ, at time *t* is represented as $Pr_{\lambda(i)|\alpha}(t)$, where *i* is an identifier value for λ. For example, considering our data set described later in the paper, α = *"MS-RDP"*, which has an associated threat λ = *"MS-RDP Brute-force attempt" (bfa)* at $t = 1$, has $Pr_{bfa|MS-RDP}(1) = 0.86$. The granularity of one time instance has been taken as an hour.

The severity of a threat targeting a software application has a corresponding numeric value ranging from 0.0 to 1.0. For ensuring that numeric values for corresponding severity levels remain equidistant from each other and within the range 0–1, the equidistant factor is calculated using $\frac{1}{l-1}$, where *l* represents the possible number of linguistic labels used for classifying the severity intensity levels. The term $l - 1$ has been used to ensure that the lowest label type always has a numeric value equal to 0.0. For example, our data set has 5 severity levels: *critical*, *high*, *medium*, *low* and *informational*. The equidistant factor for 5 severity levels is $\frac{1}{5-1} = 0.25$. The corresponding numeric values for severity labels based on the equidistant factor are: *critical* = 1.0, *high* = 0.75, *medium* = 0.50, *low* = 0.25 and *informational* = 0.0.

A constant, W_{sev}, is used to give higher preference to a threat with higher severity over another threat with a lower severity in order to avoid averaging out these varied threat severities while calculating *Risk* score. For example, α = *"Steam"* has two associated threats, λ_1 = "*WindowsDLL*" having *low* severity and λ_2 = "*7 – ZipARJFile*" with *high* severity; then W_{sev} ensures that λ_2 is given a higher weighting while calculating the risk score value for the software application, in this case *"Steam"*. W_{sev} is assigned a numeric value based on the number of severity intensity levels, *m*. The value of W_{sev} for the lowest severity level is 0.0 and increases by *m* for each higher severity level. For the aforementioned 5 severity levels, the associated constant factor values are $W_{critical} = 20.0$, $W_{high} = 15.0$, $W_{medium} = 10.0$, $W_{low} = 5.0$ and $W_{informational} = 0.0$.

Given a software application α targeted by threat λ_n at time *t*, where *n* is the threat identifier, the probability of α targeted by λ_n is $Pr_{\lambda(i)|\alpha}(t)$; severity of λ_n is *sev*; and W_{sev} is the constant factor; then we calculate *Risk score* associated with α at time *t* represented as $Risk_\alpha(t)$ using the following equation:

$$Risk_\alpha(t) = \sum_{n=1}^{j} (Pr_{\lambda(n)|\alpha}(t) * sev) * W_{sev} \quad (1)$$

For example, based on our data considering α = *"MS-RDP"* which has an associated threat λ = *"MS-RDP Brute-force attempt" (bfa)* with *sev* = *"high"*. The probability of occurrence of λ given α at $t = 1$ represented as $Pr_{bfa|MS-RDP}(1) = 0.86$. The numeric values for *sev* = *"high"* = 0.75 and $W_{high} = 15.0$ give a risk score value, $Risk_{MS-RDP}(1) = ((0.86 * 0.75) * 15.0) = 9.67$.

We further use Risk score to calculate software application-specific risk given a particular subnetwork *Z* at time *t* represented as $Risk_{\alpha|Z}(t)$ in order to identify the cyber risk hotspots due to α running in *Z*. $Pr_{\lambda(n)|\alpha,Z}(t)$ represents the probability of $\lambda(n)$ targeting α running in a subnetwork *Z* at time *t*. We use the following equation:

$$Risk_{\alpha|Z}(t) = \sum_{n=1}^{j} (Pr_{\lambda(n)|\alpha,Z}(t) * sev) * W_{sev} \quad (2)$$

Continuing from *MS-RDP* example, if *MS-RDP* is only used in Z = "*Campus*" at $t = 1$ with $Pr_{bfa|MS-RDP,Campus}(1) = 0.86$, then $Risk_{MS-RDP|Campus}(1) = 9.67$.

Risk score can be used for identifying cyber risk hotspots due to multiple software applications running in a subnetwork. A network administrator can compare the risk scores of different software applications running in a particular subnetwork at a given time to review the security policies. The results of this scenario can be used for doing a comparison between different subnetworks of a large computer network to identify the most and the least targeted subnetworks. The Risk score of a given subnetwork Z at time *t* represented as $Risk_Z(t)$ is a summation of all software application-specific risks in that particular subnetwork at that time instance (as calculated in Eq. 2). We use the following equation for calculating risk in this scenario:

$$Risk_Z(t) = \sum_{n=1}^{i} Risk_{\alpha(n)|Z}(t) \quad (3)$$

For example, in Z = "*Campus*" when three software applications, MS-RDP, web-browsing and Google-play, are used at t = 1 with $Risk_{MS-RDP|Campus}(1) = 9.67$, $Risk_{web-browsing|Campus}(1) = 1.4$ and $Risk_{google-play|Campus}(1) = 0.4$, then $Risk_{Campus}(1) = 11.47$.

3.3. *Risk grade*

As risk score (*Risk*) is often a single number (associated with λ, α or Z), it does not provide any insight about the likely causes of these threats. In Fig. 1a, the association between threats (λ) and particular software (α) which are impacted by these threats is shown – illustrating how the same threat can be found in multiple types of software applications. Conversely, in Fig. 1b, we illustrate how multiple threats (λ) may impact a particular software application (α). We define "Risk Grade" *RG* using a decision tree to help identify and rank the specific threats, λ, occurring over a time period for software α, being used in a particular subnetwork (Z). The decision tree (see Fig. 2) shows a two-level of edge hierarchy for threat classification. For example, Λ = *Spyware* will assist a network administrator to look for anti-spyware solutions. Each individual threat, λ, has an associated severity. This association between software applications and threat categories, and threat categories and threats, is represented as edges, e_i and e_j respectively. These edges form a path, *P* for each λ linked with a particular α in the decision tree.

An edge e_i represents the exploitation of a software application α running in a sub-network Z by a particular threat category, Λ. The set E_1 contains all possible e_i due to Λ_n. Similarly, edge e_j represents the specific exploitation with a specific severity *sev* by an individual threat, λ, belonging to a particular threat category, Λ. All threat-specific edges, e_j, are included in the set E_2. Then P_k is the attack path explored by a threat given a software application running in a sub-network.

Given the following formalised definition of e_i, e_j and P_k, we calculate $RG(P_k)$ using Eq. (4):

1. $e_i = (\alpha_a | Z_b, \Lambda_n) \in E_1$
2. $e_j = (\Lambda_n, \lambda_o) \in E_2$
3. $P_k = (e_i \cup e_j)$

$$RG(P_k) = (Pr(e_i) * Pr(e_j)) + W_{sev} \quad (4)$$

where $RG(P_k)$ is the Risk Grade value for attack path P_k, $Pr(e_i)$ is the probability of occurrence of a particular threat category when using a given software application, $Pr(e_j)$ is the probability of occurrence of an individual threat with a specific severity when using a given software application, and W_{sev}

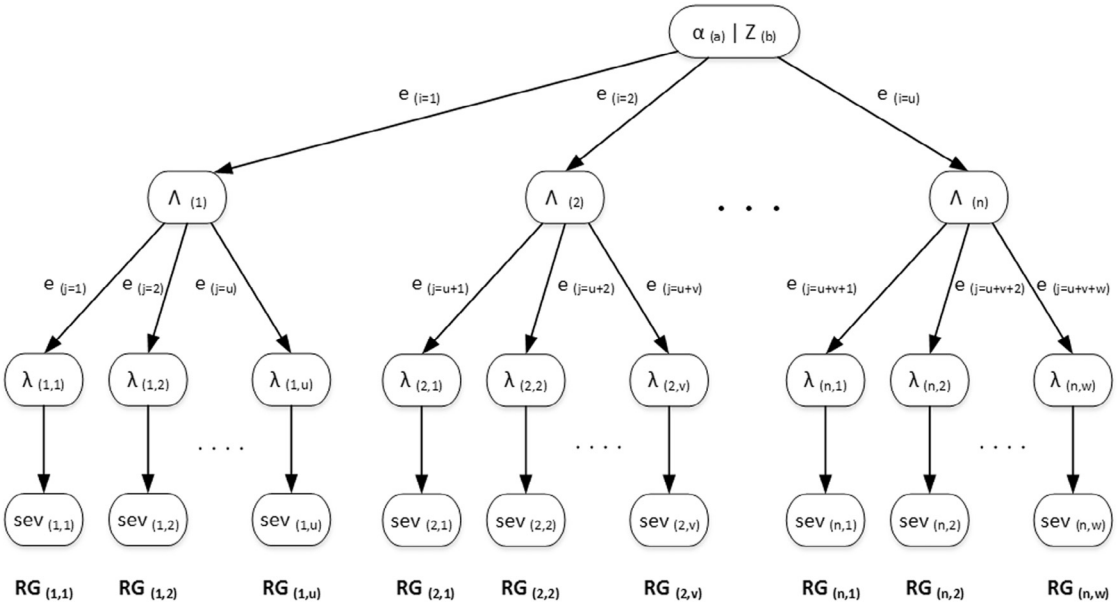


Fig. 2 – Risk grade decision tree.

is the associated weighting factor, as described in Section 3.2.

The identification of such paths enables a mitigation strategy to be supported, e.g. updating the software to a newer version, deploying more robust security software or modifying a security policy. When there are several threats at the same severity level associated with a software application in a given sub-network, *RG* value will enable a network administrator to identify how to prioritise on responding to these threats. Identification of such paths also enables threats associated with a particular software running within different subnetworks to be compared.

Given two paths, P_1 and P_2, if $RG(P_1) > RG(P_2)$ then P_1 has higher priority than P_2 to assist a network administrator take appropriate decisions for enforcing security policies.

For example, a software application, $\alpha = web-browsing$, is running in *Untrust* subnetwork (see Fig. 11). Three threat categories, *Spyware*, *Vulnerability* and *Virus*, exploit web-browsing. These three exploitation paths are represented by three edges, e_1, e_2 and e_3 respectively. The probabilities associated with each of the edges indicating possibility of following them are $Pr(e_1) = 0.0500$, $Pr(e_2) = 0.8750$ and $Pr(e_3) = 0.0750$. The threat category $\Lambda = Vulnerability$ has two threats, *HTTP related* and *ASP.NET Path*, which exploit *web-browsing* running in *Untrust* subnetwork and are represented by edges e_{j1} and e_{j2} respectively. This will give us two paths in the decision tree: (1) P_1 which consists of e_2 and e_{j1}; and (2) P_2 which consists of e_2 and e_{j2}. The probabilities of occurrences of the threats are $Pr(e_{j1}) = 0.8572$ and $Pr(e_{j2}) = 0.1428$. Both threats, *HTTP related* and *ASP.NET Path*, result in severity with a high level and have a value of $W_{high} = 15.0$. Using Eq. (6), the values of *RG* for the two paths will be: $RG(P_1) = (Pr(e_2) * Pr(e_{j1})) + W_{high} = (0.8750 * 0.8572) + 15 = 15.7501$ and $RG(P_2) = (Pr(e_2) * Pr(e_{j2})) + W_{high} = (0.8750 * 0.1428) + 15 = 15.1250$. Based on *RG* value, P_1 has higher priority than P_2.

4. Validation environment for risk framework

We have based our study on malicious network traffic data logs generated by the Palo Alto Networks IPS/IDS software Wildfire, which has been used as a security measure to protect more than 34,000 registered users of the University Campus network from cyber attacks. It protects the network from both known and previously unknown malware, zero-day exploits, and Advanced Persistent Threats (APTs). Wildfire can identify over 200 potentially malicious behaviours and is capable of classifying all traffic across nearly 400 applications (Palo Alto Networks, 2015a, 2015b). When incoming traffic is classified as malicious, it is pushed to a RabbitMQ queuing system for further investigation.

The dataset used for validating this study was collected during the regular academic year from March 26, 2014 to March 31, 2014. We accessed malicious data logs of 6 consecutive days in CSV format, obtained through RabbitMQ, for validating our risk framework. Each malicious traffic instance had 41 attributes giving information about both source and destination IP addresses, ports, zones and countries; threats, threat categories, threat severity levels, threat occurrence time; software applications targeted; classification rule, protocols, ingress and egress interfaces as well as miscellaneous information. The log files were preprocessed to extract information about threats, threat instances, threat categories, threat severity levels, threat occurrence time, software applications targeted and the subnetworks on which these threats appeared. These were then divided into hourly intervals for subsequent analysis. The collected malicious traffic data have 462,787 instances, containing 278 unique threats, across 6 distinct threat categories (vulnerability, file, virus, scan, spyware and packet). Each threat also has a severity level associated with it, and can be: informational, low, medium, high and critical. These threats originate from 86 source locations in various countries and exploit 90 different software applications used across the various subnetworks being monitored within the University networks (Campus, Campus-SrvNets, DC-DMZ, Reslan, Untrust, Wireless). Table 1 gives a description of the six subnetworks of the University networks used in this study.

Table 1 – Subnetworks in the university network.

Subnetwork	Description
Campus	This subnet contains workstations mainly used by staff and in classrooms/labs
Campus-SrvNets	This subnet provides processing of non-sensitive information as part of campus services
DC-DMZ	This subnet has physical servers providing front end/middleware services to trusted backends and is likely to be exposed to the Internet
Reslan	This subnet contains the computing devices used in student residences network
Untrust	This subnet provides the network connection routes to all sponsored sites as well as the Internet
Wireless	This subnet provides current guest roaming and extended eduroam services

Our data show 99.45% of malicious traffic targeting 14 software applications, which may be internally (within the University network) or externally hosted (accessed via a Web portal for instance). The top 7 most targeted internally hosted applications (with percentage threats over the observation period) are: (1) MS-RDP (82.18%); (2) DNS (0.85%); (3) SSH (0.78%); (4) SMTP (0.62%); (5) RPC (0.19%); (6) FTP (0.06%); and (7) MS-DS-SMB (0.04%). Likewise, the top 7 most targeted externally hosted applications/services are: (1) Web-browsing (10.78%); (2) Steam (1.17%); (3) Google-play (1.02%); (4) Sharepoint-base (0.60%); (5) Apple-Appstore (0.60%); (6) Avast-av-update (0.33%); and (7) Funshion (0.23%). These data show malicious traffic highly skewed towards the use of MS-RDP software application (based on applications that were hosted in our observed network). Our risk framework can be generalised for malicious data logs obtained from other commonly used network security software such as from IBM (IBM, 2015), McAfee (McAfee, 2015), Symantec (Symantec, 2015), Sophos (Sophos, 2015) and Zone Alarm (ZoneAlarm, 2015).

5. Results

The collected data set has been used to validate the use of our proposed risk framework, primarily to identify potential cyber

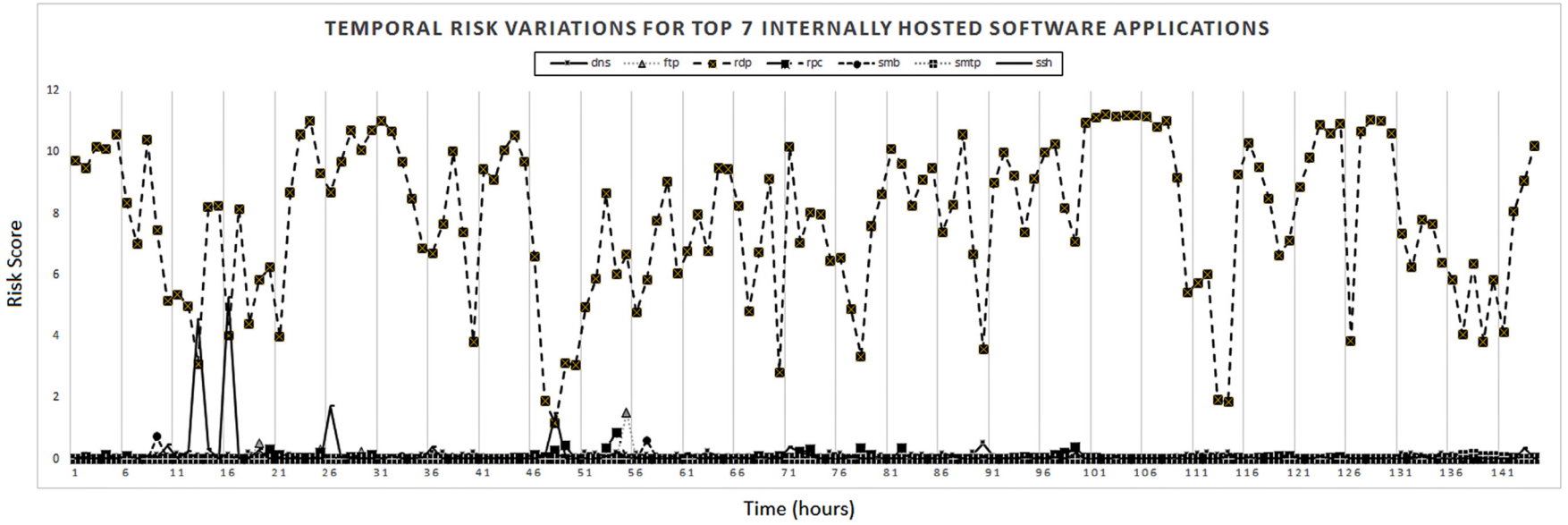


Fig. 3 – Risk score for top 7 internally hosted software applications.

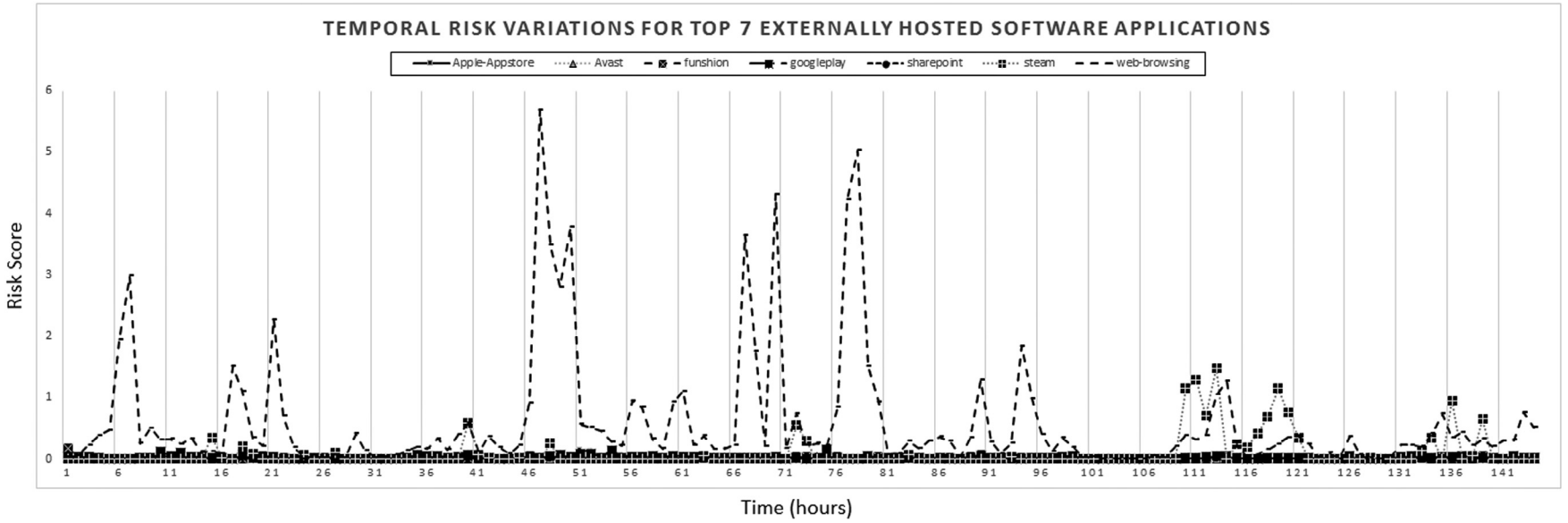


Fig. 4 – Risk score for top 7 externally hosted software applications.

risk hotspots based on the use of particular software applications. In the following subsections, we determine risk scores and how these change over our observation period for the software applications/services we monitor. Using a statistical analysis of these data, we describe how alerts can be generated for network administrators in order for them to take suitable precautionary measures. We use risk scores to identify the cyber risk hotspots emerging over the period of time and then the risk grade values for investigating the causes of emerging cyber risk hotspots.

5.1. *Risk modelling for frequently targeted software applications*

We have followed a continuous monitoring approach with a time granularity of one hour to model the risk score of the 14 most targeted software applications in the University network over a period of 144 hours. The risk score values for these most frequently targeted internally and externally hosted software applications have been calculated using Eq. (2) with Figs. 3 and 4 presenting the risk behaviour demonstrated by the top 7 internally and externally hosted software applications respectively. Table 2 shows the statistics for the modelled risk behaviour.

Based on risk occurrence statistics in Table 1, Apple-Appstore, MS-RDP and Web-browsing were targeted throughout the considered 144 hours, while SMTP was the least targeted application – only 9.72% of the 144 hours. Only 3 out of 7 internally hosted applications were targeted more than 60% of the observed time while 6 out of 7 externally hosted applications were targeted more than 83% over the observed 144 hours. Further, MS-RDP has shown the most prominent risk score variation among internally hosted software applications while Web-browsing resulted in the most prominent risk score variation among externally hosted software applications (making externally hosted applications the most vulnerable).

5.2. *Alert generation and analysis*

We suggest considering a mean risk score value, along with its probability of exceeding 1 standard deviation from mean, as an indicator for generating an alert for network administrators. Analysis of the 7 internally hosted applications (Fig. 3) revealed that the risk score values of dns, ftp, rdp, rpc, smb, smtp and ssh exceeded their mean risk score values by 38.57%, 16.21%, 55.55%, 51.66%, 47.05%, 64.28% and 21.34% respectively. The probability of risk score value exceeding more than 1 standard deviation of the mean value out of the aforementioned

Table 2 – Software application risk statistics.

Application	Host	Min.	Max.	Mean	Median	IQR	Std. dev.	Occurrence
Apple-Appstore	External	0.00	0.13	0.01	0.00	0.00–0.02	0.02	100.00%
Avast-update	External	0.00	0.04	0.00	0.00	0.00–0.01	0.00	88.88%
DNS	Internal	0.00	0.49	0.08	0.06	0.02–0.13	0.07	97.22%
FTP	Internal	0.00	1.51	0.01	0.00	0.00–0.00	0.13	25.69%
Funshion	External	0.00	0.16	0.01	0.00	0.00–0.01	0.01	93.05%
Google-play	External	0.00	0.14	0.02	0.01	0.00–0.03	0.02	98.61%
RDP	Internal	1.16	11.23	7.86	8.25	6.24–10.01	2.48	100.00%
RPC	Internal	0.00	0.86	0.04	0.00	0.00–0.04	0.10	41.66%
Sharepoint	External	0.00	0.03	0.00	0.00	0.00–0.00	0.00	83.33%
SMB	Internal	0.00	0.70	0.01	0.00	0.00–0.00	0.07	11.80%
SMTP	Internal	0.00	0.15	0.00	0.00	0.00–0.00	0.02	9.72%
SSH	Internal	0.00	5.24	0.12	0.00	0.00–0.03	0.60	61.80%
Steam	External	0.00	1.47	0.09	0.00	0.00–0.00	0.25	29.16%
Web-browser	External	0.00	5.69	0.59	0.25	0.12–0.49	0.99	100.00%

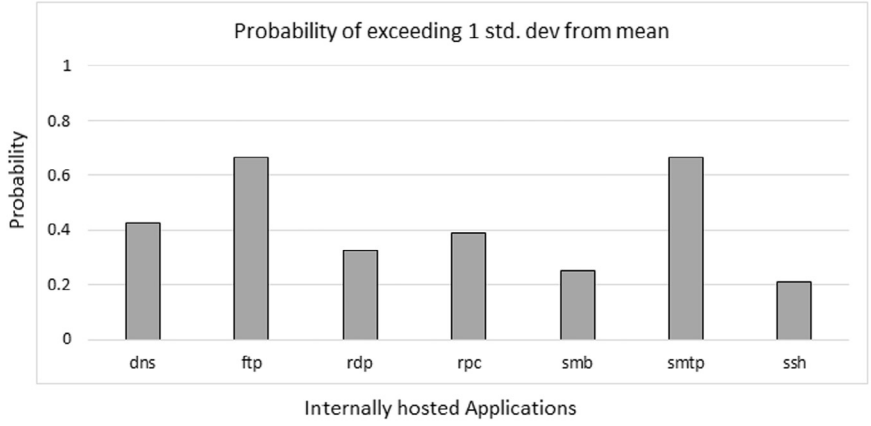


Fig. 5 – Alert generation probabilities for internally hosted applications.

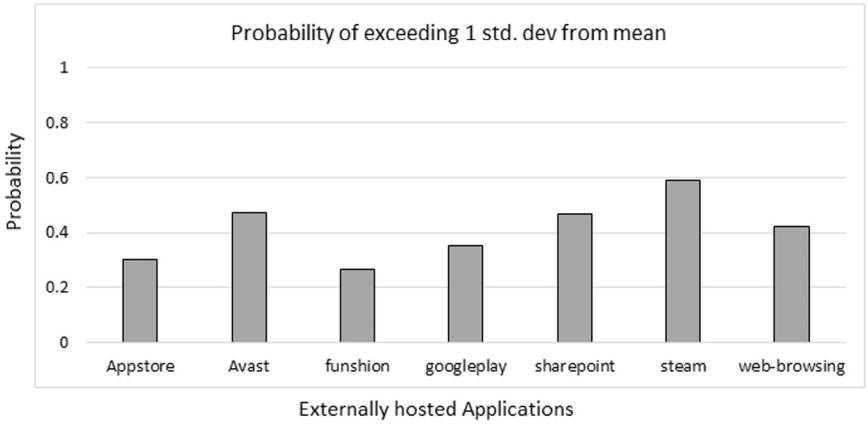


Fig. 6 – Alert generation probabilities for externally hosted applications.

Table 3 – Risk score of top software applications across the network.

Subnetwork	DNS	MSRDP	SSH	GooglePlay	Steam	Web-browsing
Campus	9.20	11.25		1.25	1.25	1.42
Campus-ServNet	–	–	11.25	–	–	9.54
DC-DMZ	5.0	–	–	–	–	8.06
Reslan	–	–	–	–	5.31	3.04
Untrust	5.0	–	–	–	–	10.43
Wireless	–	–	–	1.25	–	5.00
Network	19.20	11.25	11.25	2.50	6.56	37.49

percentage values for each of these applications (Fig. 5) suggests ssh having the lowest value with both ftp and smtp having the highest probability. A comparison with ssh suggests that ftp, smtp, dns, rpc, rdp and smb have 3.16×, 3.16×, 2.02×, 1.84×, 1.54× and 1.18× more chances than ssh to result in a risk score of exceeding more than 1 standard deviation of the mean value.

Similarly, the temporal risk behaviour analysis of 7 externally hosted applications (Fig. 4) revealed that the risk score values of Apple-Appstore, Avast-update, Funshion, Googleplay, Sharepoint, Steam and Web-browsing exceeded their mean risk score values by 34.72%, 38.58%, 36.56%, 35.91%, 39.49%, 55.00% and 21.52% respectively over the observation period. Our analysis (from Fig. 6) suggests Funshion having the lowest value with Steam having the highest risk score probability. A comparison with Funshion suggests that Steam, Avast-update, Sharepoint, Web-browsing, Google-play and Apple-Appstore have 2.22×, 1.77×, 1.76×, 1.58×, 1.33× and 1.13× more chances than Funshion to result in a risk score exceeding more than 1 standard deviation of the mean value.

We limit any additional analysis to the top 3 most targeted applications (internally or externally hosted) to identify potential risk hot spots. Table 3 presents the accumulated risk score of these applications in different subnetworks, calculated by summing up the results of Eq. (2) over the 144 hour observation period. Based on this analysis, GooglePlay has the smallest risk score among the 6 considered software applications, while the use of a Web browser has 14.99× more risk than GooglePlay – this is due to the significant use of Web-based access across various devices connected to the University network. Further, DNS, MS-RDP, SSH and Steam have 7.68×, 4.5×, 4.5× and 2.62× more risks than GooglePlay.

Fig. 7 represents the overall risk associated with different subnetworks of the University network. The circles represent the *relative risk* with respect to the lowest risk score of the Wireless subnetwork. The diamond shapes represent the software that caused risk in a particular subnetwork. Much like a crime "hotspot" map of terrestrial crime, this diagram provides an overview of the most "at-risk" sub-networks, and to what extent

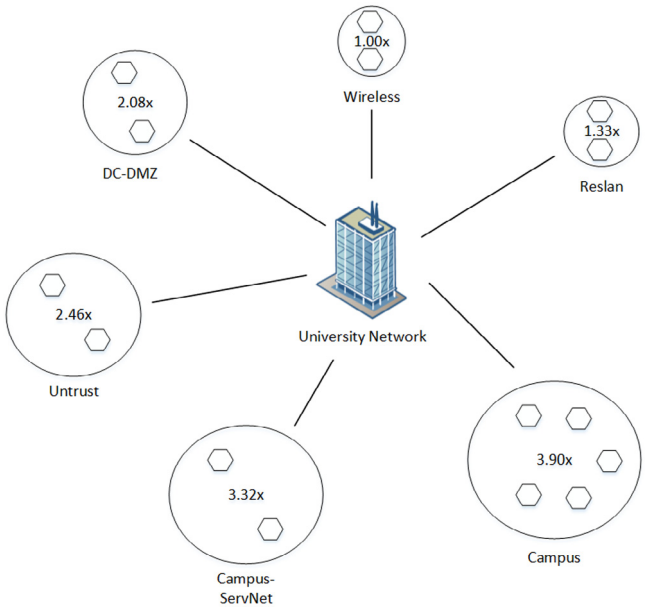


Fig. 7 – Network level risk.

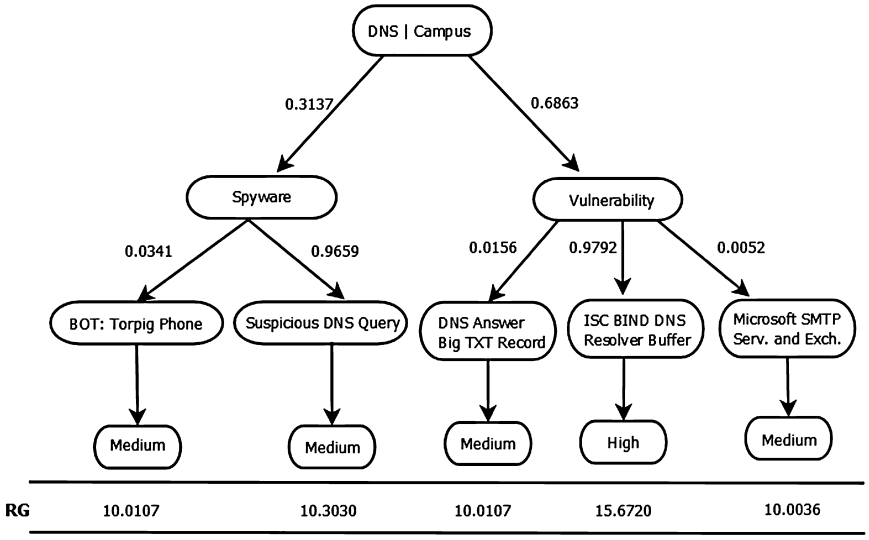


Fig. 8 – DNS threat pattern in the university campus sub-network.

particular software and services contribute to such risk. At the subnetwork level, the overall risk scores considering these 6 software applications for the *Campus*, *Campus-ServNets*, *DC-DMZ*, *Reslan*, *Untrust* and *Wireless* subnetworks are 24.37, 20.79, 13.06, 8.35, 15.43 and 6.25 respectively. These values have been calculated using Eq. (3) by summing up the individual risk score at a particular time instance. Among these sub-networks, *Wireless* has the lowest risk score classifying it as having the lowest threat level while *Campus* has 3.90× more risk than *Wireless*. Similarly, *Campus-ServNet*, *Untrust*, *DC-DMZ* and *Reslan* have higher risks than *Wireless* sub-network by 3.32×, 2.46×, 2.08× and 1.33× respectively. It is argued that the *Campus* subnetwork hosts more critical data servers and computing nodes which are used by both students and staff and is therefore more vulnerable to potential cyber-attacks. Based on our analysis, we found that MS-RDP is the main source of risk associated with *Campus*. *Wireless*, being mainly used by the visitors, has a lower risk primarily due to the limited number of potential applications that are made use of through such a network (primarily email and web-browsing).

5.3. *Modelling causes of cyber risk hotspots in software applications*

We further demonstrate how our data and risk framework can be used to investigate *causes* of cyber risk hot spots for two of most at-risk applications: DNS and Web-browsing. Moreover, DNS in *DC-DMZ* and Web-browsing in *Wireless* were targeted by a single threat – DNS ANY Queries Brute-force DOS attack and Virus/Win32.WGeneric respectively. Thus, the following subsections only compare multiple threat cases targeting the two at-risk applications.

5.3.1. *DNS*

Table 4 presents the number of distinct threats and the number of times they occur – for different threat categories targeting the DNS application. This information may be used by network administrators to improve the security policy of particular subnetworks they manage.

Fig. 8 illustrates the threat classes for the use of DNS in the *Campus* sub-net, the frequency of their occurrence, the classified severity levels based on the IDS/IPS rules and their Risk

Table 4 – DNS distinct threat count belonging to different threat categories.

Subnetwork	Vulnerability	Spyware
Campus	3	2
Untrust	1	2

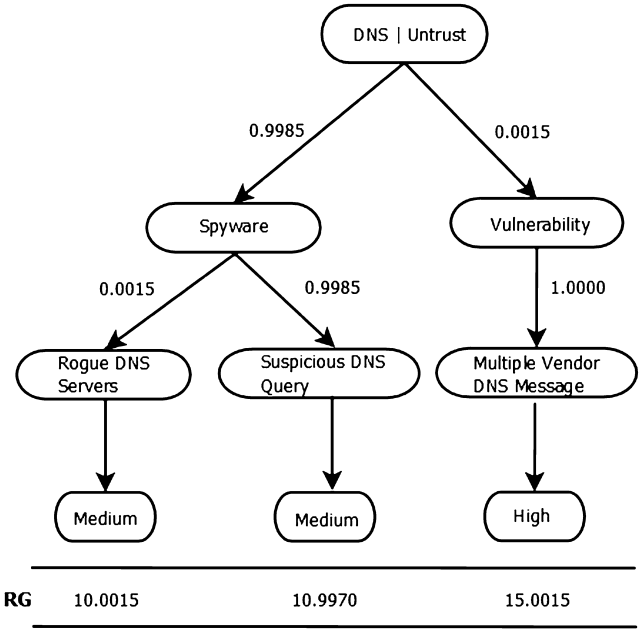


Fig. 9 – DNS threat pattern in untrust sub-network.

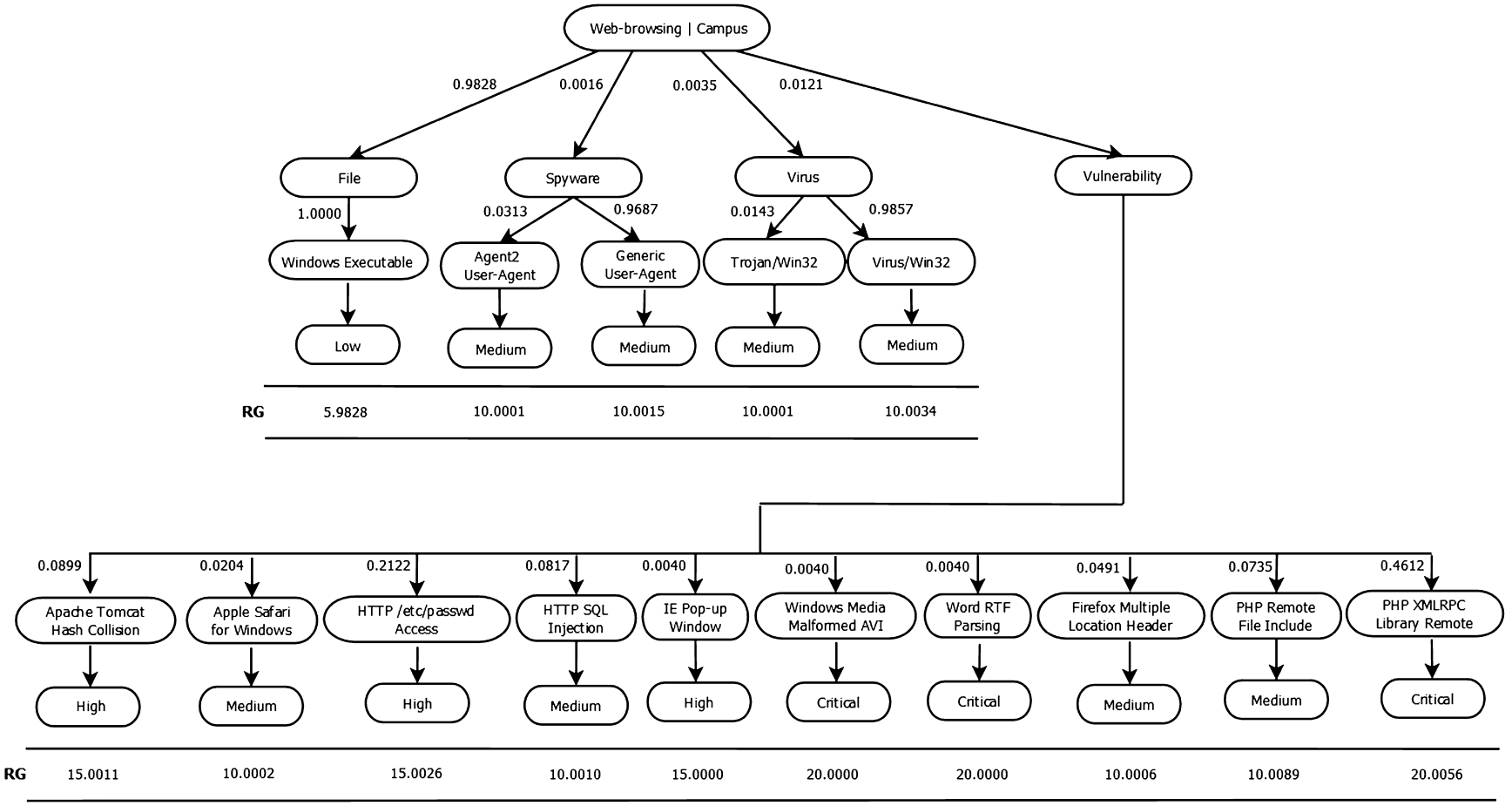


Fig. 10 – WebBrowsing threat pattern in campus.

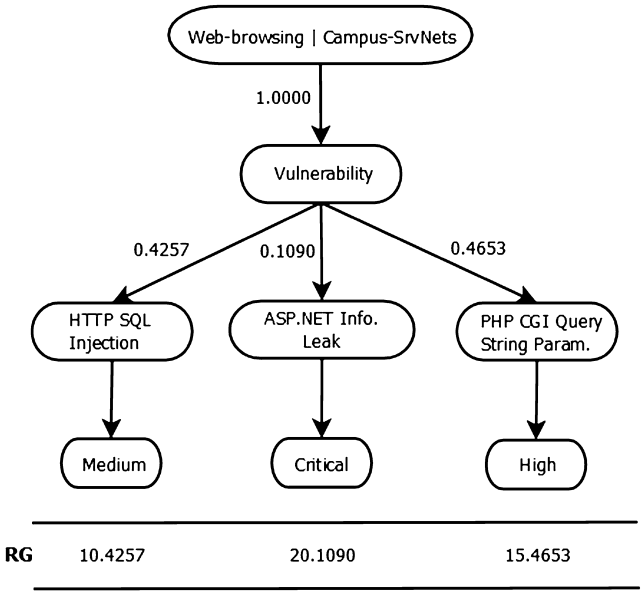


Fig. 11 – WebBrowsing threat pattern in Campus-ServNets.

Grade (RG) values. BOT:Torpig Phone, Suspicious DNS Query, DNS Answer Big TXT record and ISC BIND DNS Resolver Buffer are considered to impose a greater risk than Microsoft SMTP Server and Exchange due to their severity level and probability of occurrence – leading to a comparatively higher RG value for these. This suggests that a network administrator should prioritise on remedies for ISC BIND DNS Resolver Buffer and Suspicious DNS Query because of their higher RG values.

For comparison, Fig. 9 shows the DNS use within the *Untrust* sub-net using the same threat classes of Spyware and Vulnerability. Our dataset revealed that 95 different Suspicious DNS Queries, e.g., generic:api.megabrowse.biz, generic:api.myfindright.com, generic:s.m2pub.com, generic:check.frogupdate.com, belonging to Spyware class resulted in a

Table 5 – Web-browsing distinct threat count belonging to different threat categories.

Subnetwork	Vulnerability	Spyware	File	Virus
Campus	10	2	2	1
Campus-ServNets	3	0	0	0
DC-DMZ	6	0	0	0
Reslan	4	6	3	3
Untrust	2	1	0	1

medium level of severity/impact on the system. The "Vulnerability" threat class results in a higher severity on the network – as identified in Fig. 9. The Risk Grade values for Suspicious DNS Query and Multiple Vendor DNS Message are 1.0995× and 1.4999× more than the Rogue DNS Servers threat.

The occurrence of Suspicious DNS Query in both *Campus* and *Untrust* sub-nets suggests the use of a similar control for this threat across both. For other threats, the attack vectors targeting DNS are different in different sub-nets, requiring these threats to be considered individually rather than applying the same controls. More importantly, we can measure the effectiveness of these controls and use the outcome to update the security policy for DNS-related threats.

5.3.2. *Web-browsing*

Similar to Table 4, threat categories associated with the web-browsing application running in different sub-nets are presented in Table 5. We observe that web-browsing application running in *Campus*, *Campus-ServNets*, *DC-DMZ* and *Untrust* sub-nets exploits vulnerabilities in web-servers; in the *Reslan* sub-net Spyware is the main threat category, while in the *Wireless* sub-net the application is primarily targeted by viruses. We further use Risk Grade (RG) to determine the attacks requiring higher priority from a network administrator.

The decision tree showing the Risk Grade (RG) values for web-browsing in *Campus* is given in Fig. 10, suggesting that "PHP

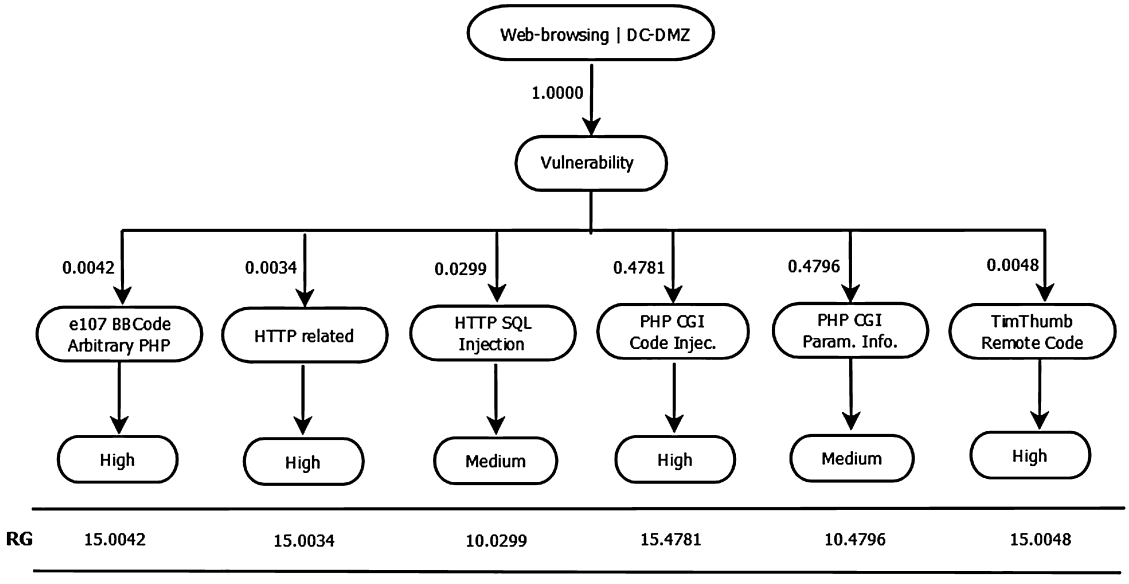


Fig. 12 – WebBrowsing threat pattern in DC-DMZ sub-net.

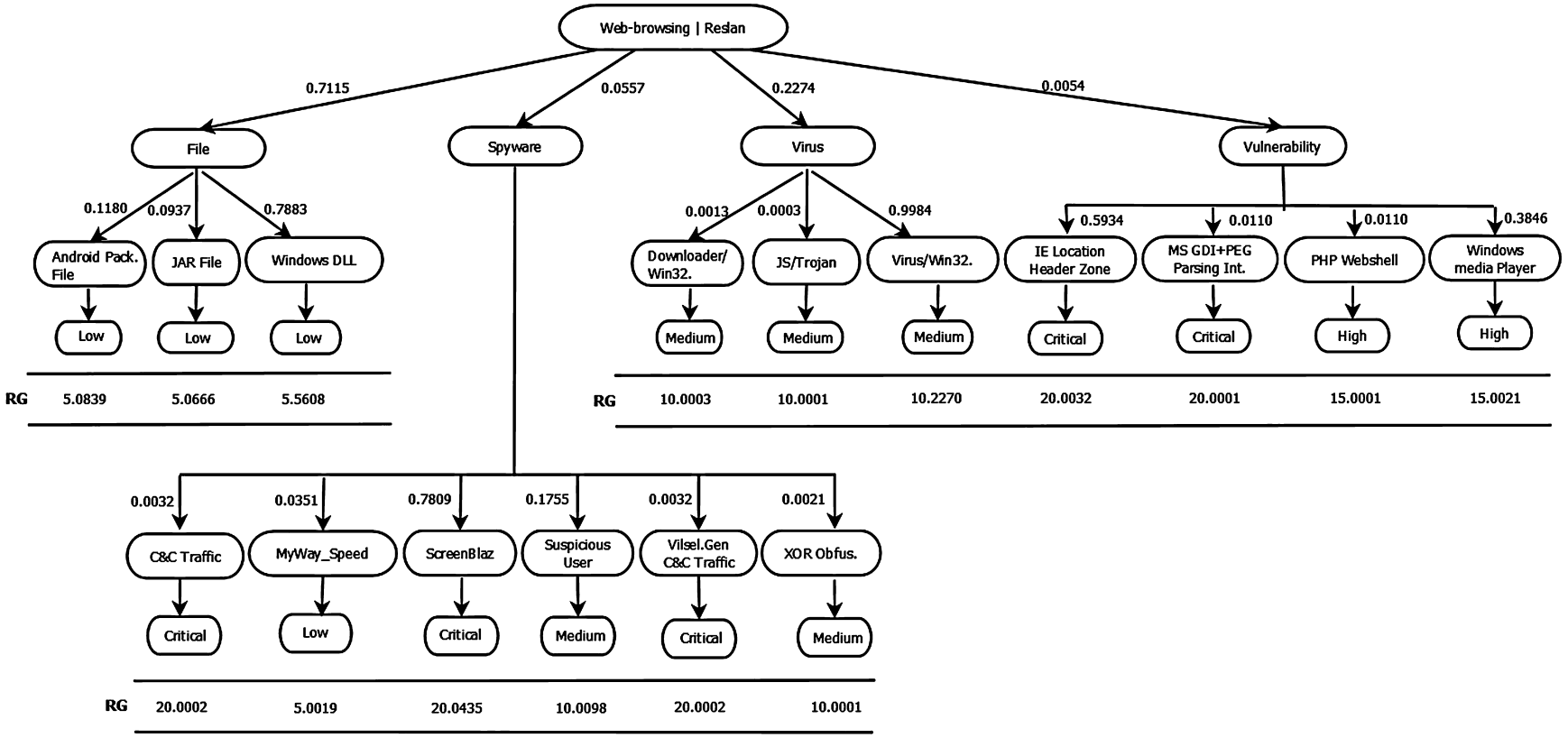


Fig. 13 – WebBrowsing threat pattern in Reslan subnet.

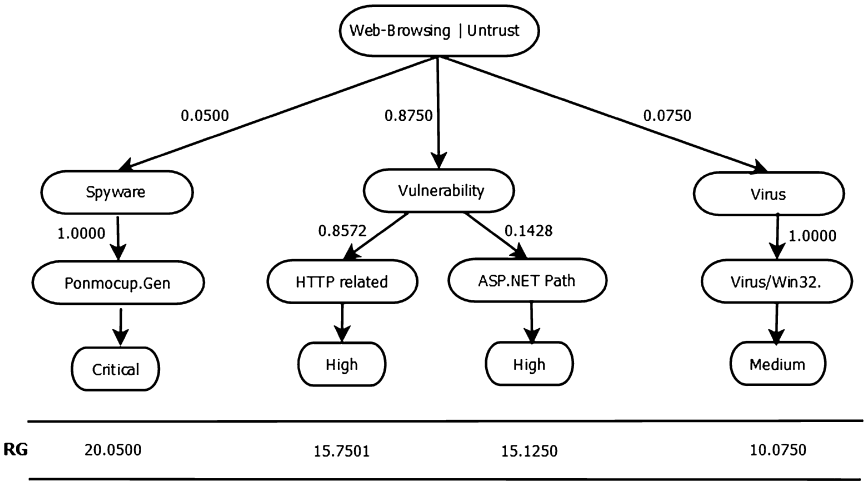


Fig. 14 – WebBrowsing threat pattern in untrust.

XMLRPX Library Remote Code Execution" vulnerability has the highest RG, followed by "MS Word RTF Parsing Engine Memory Corruption" vulnerability and "MS Windows Media Malformed AVI Parsing". These have also been identified to be "critical", suggesting urgent input from a network administrator.

In *Campus-ServNets*, web-browsing has three threats in the "Vulnerability" class with varying levels of severity (see Fig. 11), with "ASP.NET Info. Leak" dominating (based on its Risk Grade value) and identified as being "critical". In the *DC-DMZ* sub-net (see Fig. 12), "PHP CGI Code Injection threat" has the highest severity – indicating greater required input from a network administrator.

Like *Campus*, web-browsing in *Reslan* sub-net (see Fig. 13) has four threat categories associated with it, namely: File, Spyware, Virus and Vulnerability. According to the Risk Grade values and severity levels, the greatest threats occur in the "Spyware" and "Vulnerability" categories. Both of these require additional support from a network administrator to handle these threats – the highest being the "ScreenBlaz" spyware – a malicious Trojan virus that has been the main cause of browser hijacking in Win32 systems.

Three threat categories, Spyware, Virus and Vulnerability, were observed in *Untrust* sub-net when using web-browsing service (see Fig. 14). The Risk Grade values for Ponmocup.Gen C&C, HTTP related and ASP.NET Path are more than Virus/Win32 threat by 1.9900×, 1.5632× and 1.5012× respectively suggesting that a network administrator should first find a response for Ponmocup.Gen C&C.

Comparison across these suggests that "HTTP SQL Injection" is a common threat occurring in Campus, Campus-ServNets and DC-DMZ sub-nets. Similarly, HTTP related threats are common in DC-DMZ and Untrust sub-nets. This suggests that: (i) a greater focus should be considered on such commonly occurring threats; (ii) controls used for this threat can be shared across multiple sub-nets. The treatment of remaining threats requires individual actions on the part of network administrator.

6. Discussion and conclusion

In this paper we developed a risk assessment framework that we propose could be used by network administrators and people responsible for managing network security risk to take a high-level view of the overall network and immediately identify sub-nets at risk to quickly identify the cause of the risk, taking remedial action to rectify the problem. We demonstrated the application of the framework using real-world data collected from a Local Area Network to Internet gateway.

It was evident within the results that externally hosted software applications, such as Steam and Google-play, are leading to threats from malicious files, which require a different type of threat management from the internally hosted software applications, such as DNS, where the software application itself is proving to the key vulnerability. This is a reflection of the new type of risk evolving from Cloud services and ad-hoc devices downloading and running applications on the network. The key threats are coming from outside and being intentionally pulled into the network, rather than attackers having to hack their way into the system. The ad-hoc use of personal devices within corporate networks could exacerbate this problem, and we have developed the framework so that it can identify software or application-specific risks within a sub-net to support the management and identification of emerging risks due to the range of "apps".

We have provided a snap shot of several days and shown how the most pertinent threats fluctuate and alter over time. This is an important finding as it suggests a static security solution that aims to mitigate risk at a network level is not as suitable as a risk management model that can be dynamic and reactive to emerging risks at a sub-net level. The temporal aspect of the framework is flexible and could be reduced to a few hours or many months, providing shorter and longer views on risk over time. We suggest that, if this kind of activity were to take place periodically, it would allow network security policies to be updated and enforced more effectively and efficiently,

reducing harm to the network and keeping in line with the current modus-operandi of cyber criminal network behaviour. With so many risk assessment framework standards emerging, it is important to remember that risk assessment in Cyber security must occur frequently and remain an active process – even after the standard processes of asset identification, threat and vulnerability assessment, and impact analysis are complete.

Our results use a single local area network to demonstrate the framework. Although it is a large network with a variety of use cases, from personal to corporate to scientific research, we cannot yet claim that the proposed framework would be generalisable to finance, retail, and industry control systems or healthcare settings. Thus, we suggest that this work provide a foundation and clarion call for further research in this area and the publication of findings across a variety of contexts to provide comparable results from which to build evidence of a generalisable real-time risk assessment framework.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council Global Uncertainties Consortia for Exploratory Research in Security (CEReS) programme, grant number: EP/K03345X/1.

REFERENCES

Burnap P, Hilton J. Self protecting data for de-perimeterised information sharing. Proc. 3rd IEEE Int'l Conf. Digital Society. ICDS 2009. Cancun, Mexico. 2009.

Dantu R, Loper K, Kolan P. Risk management using behavior based attack graphs, Proc. Int'l Conf. Information Technology: Coding and Computing, pp. 445–9, 2004.

Dantu R, Kolan P, Akl R, Loper K. Classification of attributes and behavior in risk management using Bayesian networks, Proc. IEEE Intelligence and Security Informatics Conf., pp. 71–4, 2007.

Dantu R, Kolan P, Cangussu J. Network risk management using attacker profiling. Secur Commun Netw 2009;2:83–96.

First. Common Vulnerability Scoring System (CVSS-SIG), <http://www.first.org/cvss>; 2015 [accessed 19.08.14].

Frigault M, Wang L, Singhal A, Jajodia S. Measuring network security using dynamic Bayesian network, Proc. 14th ACM Workshop Quality of Protection, 2008.

IBM. IBM Security Network Intrusion Prevention System, <http://www-03.ibm.com/software/products/en/network-ips>; 2015 [accessed 23.12.14].

ISO. Information technology – Security techniques – Information security management systems – Requirements, ISO/IEC 27001, <http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54534>; 2013 [accessed 19.08.14].

Kanoun W, Cuppens-Boulahia N, Cuppens F, Araujo J. Automated reaction based on risk analysis and attackers skills in intrusion detection systems, Proc. of Third International Conference on Risks and Security of Internet and Systems, 2008. pp. 117–24.

Kheir N, et al. A service dependency model for cost sensitive intrusion response, Proc. of the 15th European Conference on Research in Computer Security; 2010. pp. 626–42.

Kott A, Arnold C. The promises and challenges of continuous monitoring and risk scoring. IEEE Secur Priv 2013;11(1):90–3.

Liu Y, Man H. Network vulnerability assessment using Bayesian networks. Proc SPIE 2005;5812:61–71.

Lund MS, Solhaug B, Stolen K. Model-driven risk analysis: the CORAS approach. Springer; 2011.

McAfee. McAfee – Antivirus, Encryption, Firewall, Email Security, Web Security, Risk & Compliance, <http://www.mcafee.com/uk/>; 2015 [accessed 23.12.14].

Mu CP, Li Y. An intrusion response decision-making model based on hierarchical task network planning. Expert Syst Appl 2010;37(3):2465–72.

Palo Alto Networks. Network Security – Next Generation Firewalls by Palo Alto Networks, <https://www.paloaltonetworks.com>; 2015a [accessed 23.12.14].

Palo Alto Networks. Palo Alto Networks: Wildfire datasheet, <https://www.paloaltonetworks.com/content/dam/paloaltonetworks-com/en_US/assets/pdf/datasheets/wildfire/wildfire.pdf>; 2015b [accessed 18.03.15].

Poolsappasit N, Dewri R, Ray I. Dynamic security risk management using Bayesian attack graphs. IEEE Trans Dependable Secure Comput 2012;9(1):61–74.

Shameli-Sendi A, Cheriet M, Hamou-Lhadj A. Taxonomy of intrusion risk assessment and response system. Comput Secur 2014;45:1–16.

Sophos. Cloud Antivirus, Endpoint, UTM, Encryption, Mobile, DLP, Server, Web, Wireless Security, Network Storage and Next-Gen Firewall Solutions |Sophos Data Protection for Business, <http://www.sophos.com/en-us.aspx>; 2015 [accessed 23.12.14].

Symantec. Endpoint, Cloud, Mobile & Virtual Security Solutions |Symantec, <http://www.symantec.com/index.jsp>; 2015 [accessed 23.12.14].

Wang S, Zhang Z, Kadobayashi Y. Exploring attack graph for cost-benefit security hardening: a probabilistic approach. Comput Secur 2013;32:158–69.

Xie P, Li JH, Ou X, Liu P, Levy R. Using Bayesian Networks for Cyber Security Analysis, Proc. 40th IEEE/IFIP Int'l Conf. Dependable Systems and Networks, 2010.

Zhang Z, Ho PH, He L. Measuring IDS-estimated attack impacts for rational incident response: a decision theoretic approach. Comput Secur 2009;28(7):605–14.

ZoneAlarm. Firewall Antivirus Software and Computer Security Suite by ZoneAlarm, <http://www.zonealarm.co.uk>; 2015 [accessed 23.12.14].

Malik Shahzad Kaleem Awan is a research associate in the School of Computer Science & Informatics at Cardiff University, working in the area of cybersecurity and distributed systems. He holds a PhD in Computer Science from the University of Warwick, UK.

Peter Burnap is an assistant professor/lecturer in the School of Computer Science & Informatics at Cardiff University. He is an interdisciplinary researcher, working closely with the Cardiff School of Social Sciences and School of Engineering. His research focus is cyber conflict, crime and security, more specifically, the analysis and understanding of online human and software behaviour, with a particular interest in emerging and future risks posed to civil society, business (economies) and governments, using computational methods such as machine learning and statistical data modelling, and interaction and behaviour mining, opinion mining and sentiment analysis to derive key features of interest. He holds a PhD in Computer Science from Cardiff University, UK.

Omer Rana is a professor of Performance Engineering in the School of Computer Science & Informatics at Cardiff University. His research interests include high performance distributed computing, data mining/analysis and multiagent systems. He holds a PhD in Computer Science from Imperial College, London, UK.